Computers & Industrial Engineering 107 (2017) 276–279

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

On exact algorithms for single-machine scheduling problems with a variable maintenance

Qi Wang *, Aihua Liu, Junfang Xiao

School of Science, East China University of Technology, Nanchang, Jiangxi 330013, China

ARTICLE INFO

Article history:
Received 8 November 2016
Received in revised form 13 February 2017
Accepted 20 March 2017
Available online 21 March 2017

Keywords:
Scheduling
Variable maintenance
Exact algorithms

ABSTRACT

The aim of this paper is to point out that all mathematical programming models proposed by Ying et al. (2016) are incorrect. We present four revised mathematical programming models and four improved mathematical programming models by adding and revising some constraints and decision variables. Moreover, we show that the first three scheduling problems considered in their paper are equivalent to the problems with the objective of minimizing the sum of completion times or minimizing the maximum lateness, which can be solved by algorithms proposed by Luo et al. (2015) in $O(n^2)$ time.

1. Introduction

In a recent paper, Ying, Lu, and Chen (2016) consider four single-machine scheduling problems with a variable maintenance (VM), where the machine is continuously available but must undertake a maintenance activity during the planning horizon and the duration of maintenance is a positive and non-decreasing function of its start time.

Following the same notations and terminologies in Ying et al. (2016), the scheduling scenario under consideration can be described as follows. There is a set of jobs $J = \{1, 2, \ldots, n\}$ ready to be processed on a single machine with a maintenance activity. All jobs are non-resumable and simultaneously available for processing at the beginning of the planning horizon. The processing time of job j is p_j. The start time of the maintenance activity, s, is a decision variable and must be before a given deadline s_d where $s_d < \sum_{j=1}^{n} p_j$. The duration of the maintenance activity, l, is a positive and non-decreasing function of its start time s, i.e., $l = f(s)$. The machine cannot process any job during the period of the maintenance activity.

Ying et al. (2016) consider four objectives, i.e., minimizing mean lateness, minimizing maximum tardiness, minimizing total flow time, and minimizing mean tardiness, each at a time, under the above scheduling scenario. In other words, they consider the following four scheduling problems: $1, VM||\overline{L}, 1, VM||T_{\max}, 1, VM|r_j = r|\sum_j F_j$, and $1, VM|d_j = d|\overline{T}$, where r_j and d_j are the release time and due-date of job j, respectively. Let C_j be the completion time of job j. Then $F_j = C_j - r_j, L_j = C_j - d_j$ and $T_j = \max\{0, L_j\}$. And then $\overline{L} = \sum_{j=1}^{n} L_j \big/ n, T_{\max} = \max\{T_1, T_2, \ldots, T_n\}, L_{\max} = \max\{L_1, L_2, \ldots, L_n\}, \overline{T} = \sum_{j=1}^{n} T_j \big/ n$. For each of the four problems, Ying et al. (2016) propose a mixed integer linear programming model and provide a polynomial-time optimal algorithm.

As we observe, all the mixed integer linear programming models they proposed are incorrect and the first three scheduling problems considered in their paper are equivalent to the problems with the objective of minimizing the sum of completion times or minimizing the maximum lateness, which can be solved by algorithms proposed by Luo, Cheng, and Ji (2015) in $O(n^2)$ time.

For convenience, denote by $J_{[i]}$ the job at the ith processing position. Let $d_{[k]}, r_{[k]}$ and $C_{[k]}$ be the due date, the release time and the completion time of $J_{[i]}$, respectively. Denote by π^i the candidate schedule with the maintenance activity arranged before the start time of $J_{[i]}$. For candidate schedule π^1, the maintenance activity executed at time zero. Let $C_{[k]}(\pi^i), f(C_{[i-1]}), T_{\max}(\pi^i)$ and $\overline{T}(\pi^i)$ be the completion time of $J_{[k]}$, the duration of maintenance activity, the maximum tardiness of the jobs and the mean tardiness of the jobs in candidate schedule π^i, respectively.

2. On the mathematical programming models

Let x_{ij} be binary decision variables. If job j is assigned to position i, then $x_{ij} = 1$, and $x_{ij} = 0$ otherwise.

* Corresponding author.
E-mail addresses: qwang@ecut.edu.cn, wang_qi96@163.com (Q. Wang), ahliu@ecut.edu.cn (A. Liu), jfxiao@ecut.edu.cn (J. Xiao).

http://dx.doi.org/10.1016/j.cie.2017.03.024
0360-8352/

2.1. Models proposed by Ying, Lu and Chen

For problem $1, VM\|\bar{L}$, Ying et al. (2016) propose the following model.

Minimize z

subject to

$$z \geqslant \sum_{k=1}^{n}(C_{[k]}(\pi^i) - d_{[k]})/n, \quad i = 1,2,\ldots,n, \tag{1}$$

$$C_{[k]}(\pi^i) = \sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij}p_j, \quad k = 1,2,\ldots,i-1, \tag{2}$$

$$C_{[k]}(\pi^i) = \sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij}p_j + f(C_{[i-1]}), \quad k = i, i+1\ldots,n, \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n, \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n, \tag{5}$$

$$x_{ij} \geq 0. \tag{6}$$

For problem $1, VM\|T_{\max}$, Ying et al. (2016) propose the following model.

Minimize z

subject to

$$z \geqslant T_{\max}(\pi^i), \quad i = 1,2,\ldots,n, \tag{7}$$

$$T_{\max}(\pi^i) \geqslant C_{[k]}(\pi^i) - d_{[k]}, \quad k = 1,2,\ldots,n, \tag{8}$$

$$T_{\max}(\pi^i) \geq 0, \tag{9}$$

constraints $(2) - -(6)$.

For problem $1, VM|r_j = r|\sum_j F_j$, Ying et al. (2016) propose the following model.

Minimize z

subject to

$$z \geqslant \sum_{k=1}^{n}(C_{[k]}(\pi^i) - r_{[k]}), \quad i = 1,2,\ldots,n, \tag{10}$$

constraints $(2) - -(6)$.

For problem $1, VM|d_j = d|\bar{T}$, Ying et al. (2016) propose the following model.

Minimize z

subject to

$$z \geqslant \sum_{k=1}^{n} T_{[k]}(\pi^i)/n, \quad i = 1,2,\ldots,n, \tag{11}$$

$$T_{[k]}(\pi^i) \geqslant C_{[k]}(\pi^i) - d_{[k]}, \quad k = 1,2,\ldots,n, \tag{12}$$

$$T_{[k]}(\pi^i) \geqslant 0, \quad k = 1,2,\ldots,n, \tag{13}$$

constraints $(2) - -(6)$.

2.2. Comments on the models

Ying et al. (2016) call these models mixed integer *linear* programming mathematical models. However, as we do not know whether f is linear or not, so such a name is *not* appropriate here.

As we observe, there are some fatal errors in the models. First of all, the aims of these models are seeking for the worst rather than the best (optimal) schedules. Taking the first model for instance, according to constraint set (1), z must be larger than or equal to the mean lateness of any candidate schedule. Thus, minimizing z is equivalent to force the variable to take the objective value of the worst candidate schedule. We will see this by solving a scheduling instance at the end of this subsection.

Secondly, there are some constraints which are missing in Ying et al. (2016). Note that the start time of the maintenance activity must be before a given deadline s_d, so we have

$$C_{[i-1]}(\pi^i) \leqslant s_d, \quad i = 1,2,\ldots,n, \tag{14}$$

and

$$C_{[0]}(\pi^i) = 0, \quad i = 1,2,\ldots,n. \tag{15}$$

Note that $d_{[k]}$ represents the due date of $J_{[k]}$ in candidate schedule π^i in problem $1, VM\|\bar{L}$, problem $1, VM\|T_{\max}$ and problem $1, VM\|\bar{T}$. So we need to add the following constraint set in the mathematical programming models for these problems.

$$d_{[k]} = \sum_{j=1}^{n} x_{kj}d_j, \quad k = 1,2,\ldots,n. \tag{16}$$

Moreover, noting that x_{ij} are binary variables, constraint set (6) should be

$$x_{ij} \in \{0,1\}, \quad i,j = 1,2,\ldots,n. \tag{17}$$

Even we add these missing or revised constraint sets to the corresponding models, they are still unable to generate the desired solutions. Let us take the model for the first scheduling problem as an example. Consider the following instance: $J = \{1,2\}, n = 2, p_1 = 2, p_2 = 3, d_1 = 1, d_2 = 1, s_d = 4, f(x) = 1 + x$.

It is easy to see that there are only two candidate schedules for this instance. For problem $1, VM\|\bar{L}$, solving the model, we obtain $x_{11} = x_{22} = 1, x_{12} = x_{21} = 0$. In other words, the model generates candidate schedule $\pi^2 = (1, MA, 2)$ with an objective value of 4. However, one may find that the other candidate schedule, $\pi^1 = (MA, 1, 2)$, is optimal with an objective value of 3.5, which is less than that of the generated candidate schedule π^2 by the model. This indicates that the model is seeking for the worst rather than the best (optimal) schedule.

2.3. The revised models

Recall that π^i is a candidate schedule with the maintenance activity arranged before job $J_{[i]}$. Generally speaking, there are $n!$ *feasible* schedules where the maintenance activity is arranged before the ith position of a processing sequence. Obviously, the candidate schedule π^i is the best one among them. And for this reason one cannot revise these models by maximizing z with constraint set (1) in reverse order (i.e., replacing $\geqslant$ with $\leqslant$ in constraint set (1)) to get an optimal schedule.

Assume that $p_{l_1} \leqslant p_{l_2} \leqslant \cdots \leqslant p_{l_n}$, where $(l_1, l_2, \ldots, l_n)$ is a permutation of $(1,2,\ldots,n)$. Let k^* be an integer such that $\sum_{i=1}^{k} p_{l_i} \leqslant s_d < \sum_{i=1}^{k+1} p_{l_i}$. Clearly, there are k^* candidate schedules. If we can determine all the candidate schedules, then we can obtain an optimal schedule easily by choosing the best one from them.